ADVANCED ENGINEERING MATERIALS
www.aem-journal.com

FAIR and Structured Data: A Domain Ontology Aligned with Standard-Compliant Tensile Testing

Markus Schilling, Bernd Bayerlein, Philipp von Hartrott, Jörg Waitelonis, Henk Birkholz, Pedro Dolabella Portella, and Birgit Skrotzki*

The digitalization of materials science and engineering (MSE) is currently leading to remarkable advancements in materials research, design, and optimization, fueled by computer-driven simulations, artificial intelligence, and machine learning. While these developments promise to accelerate materials innovation, challenges in quality assurance, data interoperability, and data management have to be addressed. In response, the adoption of semantic web technologies has emerged as a powerful solution in MSE. Ontologies provide structured and machine-actionable knowledge representations that enable data integration, harmonization, and improved research collaboration. This study focuses on the tensile test ontology (TTO), which semantically represents the mechanical tensile test method and is developed within the project Plattform MaterialDigital (PMD) in connection with the PMD Core Ontology. Based on ISO 6892-1, the test standard-compliant TTO offers a structured vocabulary for tensile test data, ensuring data interoperability, transparency, and reproducibility. By categorizing measurement data and metadata, it facilitates comprehensive data analysis, interpretation, and systematic search in databases. The path from developing an ontology in accordance with an associated test standard, converting selected tensile test data into the interoperable resource description framework format, up to connecting the ontology and data is presented. Such a semantic connection using a data mapping procedure leads to an enhanced ability of querying. The TTO provides a valuable resource for materials researchers and engineers, promoting data and metadata standardization and sharing. Its usage ensures the generation of finable, accessible, interoperable, and reusable data while maintaining both human and machine actionability.

1. Digitalizing Materials Science and Engineering

Materials science and engineering (MSE) is undergoing a major shift of paradigms toward more efficient digitalization. Integration and reuse of data and knowledge from material synthesis, production, characterization, and modeling activities open new perspectives for innovation. Emerging fields of materials informatics employing tools such as machine learning (ML), big-data applications, statistical inference, and integrated computational materials engineering allow the discovery of new compositions and processes tailored to produce materials with specific microstructures and properties. These advancements not only drive materials innovation but also lead to improved performance in various applications. Efficient modeling and simulation of materials engineering processes is based on large amounts of heterogeneous experimental and simulation data. These data capture multiple scales of magnitude and the diversity of relevant physical concepts such as thermodynamics, kinetics, functional, and mechanical properties as well as metadata on materials history, data origin, provenance, and environmental impacts. Hence, the innovative development, design, and optimization of materials is intrinsically linked to the digitalization of materials and processes, whereby the general ongoing advancement in hardware and software over the last decades fosters the development of materials enormously.

M. Schilling, B. Bayerlein, B. Skrotzki
Department 5.2 - Metallic High-Temperature Materials
Bundesanstalt für Materialforschung und -prüfung (BAM)
Unter den Eichen 87, 12205 Berlin, Germany
E-mail: markus.schilling@bam.de

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/adem.202400138.

DOI: 10.1002/adem.202400138

P. von Hartrott, P. Dolabella Portella
Assessment of Materials and Lifetime Concepts
Fraunhofer-Institut für Werkstoffmechanik IWM
Wöhlerstr. 11, 79108 Freiburg im Breisgau, Germany

J. Waitelonis
Information Service Engineering
FIZ Karlsruhe - Leibniz Institute for Information Infrastructure
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany

H. Birkholz
Process Engineering
Leibniz-Institut für Werkstofforientierte Technologien – IWT
Badgasteiner Str. 3, 28359 Bremen, Germany

ADVANCED ENGINEERING MATERIALS
www.aem-journal.com

The creation of interoperable digital representations of materials and processes associated with this paradigm shift, enabling a wide range of opportunities by leveraging the advantages of the digital age, coincidentally imposes a major challenge on researchers, processes, and techniques.[1–3] In particular, digitalization efforts are supposed to provide quality assurance of processes and output data as well as interoperability between applications and data. This includes storage, processing, and retrieval of data in a preferably standardized form, while also addressing the incorporation of standardization bodies. Furthermore, an appropriate management of materials data requires the adoption of findable, accessible, interoperable, and reusable (FAIR) principles.[4] To succeed in the challenge of contextualizing materials data in a way that is consistent with all stakeholders and meeting the requirements of the FAIR principles, all required information on the condition of the material including production and application-related changes has to be made available via a uniform, machine-readable, and actionable description. For this purpose, semantic web technologies (SWT) have emerged to be used, as they enable both human-readable and machine-actionable and -interpretable knowledge representations through semantic conceptualizations. Being an integral part thereof, ontologies are increasingly established as a resilient tool in the world of MSE for the implementation of complex data management and representing a basis for an information ecosystem. They are essential for formally representing universal materials science concepts, their interrelationships, and workflows. In this context, an ontology is a formally ordered and explicitly described representation of a set of terms ("concepts") from a particular area of subject ("domain") and the relationships ("properties") existing among them. The terms used are selected based on a consensus within the domain under consideration (shared, unified vocabulary) and are always supposed to be defined in natural language to facilitate human readability. Thus, the creation of an ontology results in a structure of knowledge that goes beyond a hierarchical, taxonomic description and can represent an extensive network of information with logical relations. The explicit and basic description of the concepts and relations allows the application of description logics (DL, cf. Section 3.2). In this way, both general knowledge and knowledge related to specific topics and processes can be exchanged between digital services and applications in a uniform manner.

With respect to the currently growing topic of integration and reuse of data and knowledge from synthesis, production, and characterization of materials, several national and international initiatives and projects such as the Materials Genome Initiative,[5] the DICE Materials Data Platform,[6] the Diadem materials exploratory,[7] materplat,[8] Materials Commons,[9] nanomaterial registry,[10] the Building the Prototype Open Knowledge Network program,[11] as well as the European Materials Modelling Council[12] and the National Research Data Infrastructure (NFDI)[13] with the specific MSE-related consortium NFDI-MatWerk[14] pursue the goal to develop and establish ontologies for the MSE domain, to define shared data formats, and to provide collaboration platforms as well as publicly available repositories on materials data to generate best possible benefits from materials digitalization.

This study presents the efforts within the collaborative project Platform MaterialDigital (PMD)[15] to store materials characterization data in accordance with a test standard-compliant ontological representation. Therefore, the tensile test of metals at room temperature according to ISO standard 6892-1:2019-11[16] was selected to be semantically described. This includes following the path from the development of an ontology compatible with a test standard to the conversion of common and arbitrarily selected standard test data into an interoperable data format. Finally, the correspondingly structured data can be stored in an openly accessible database to be queried by humans and machines.

This study aims to provide a best practice on an effective and straightforward approach to creating an ontology, exemplified by the semantic representation of the tensile test on metals at room temperature, which supports the MSE community in developing ontologies for their own experiments and processes. Furthermore, the tensile test ontology (TTO) is introduced which is openly available and can be used by the community.

2. Mechanical Testing

Mechanical testing of materials is a crucial aspect of MSE, serving as a fundamental method for understanding and characterizing the mechanical behavior of materials under different conditions. General aspects on mechanical testing, the development and increasing usage of digitalization within the field of tensile testing, and modern data acquisition applying electronic laboratory notebooks (ELN) are considered in the following sections.

2.1. General Aspects

One primary objective of mechanical testing is to obtain reproducible and reliable results. Reproducibility ensures that the test outcomes can be consistently duplicated under equal conditions, while reliability ensures that the results accurately reflect the mechanical properties of materials. Standardization plays a pivotal role in mechanical testing. International standards established by standardization bodies such as DIN, ASTM, and ISO as well as standard operating procedures of companies and institutes provide guidelines for conducting tests, specifying procedures, equipment, and the contents to be reported to obtain universally agreed testing methodologies. Standards ensure consistency and quality in mechanical testing by defining the scope and purpose of tests including the types of materials, components, or products covered as well as the intended applications and industries. Requirements for the testing equipment including considerations for calibration, accuracy, precision, capacities, control units, environmental conditions are specified as well as parameters with respect to sample preparation, geometry, and testing conditions.

Since the engineering community invests significantly in generating valuable materials test data, it should be well structured for comprehensive reuse. However, there is potential improvement concerning aspects of data capture, preservation, and sharing. Therefore, data management and handling techniques, such as SWT, ontologies, and data schema, are increasingly explored for usage in mechanical testing (see Section 3). They are supposed to complement existing test specifications and corresponding materials testing standards, respectively.[17] Accordingly, there is a growing emphasis on data standardization, interoperability, and automation in modern materials research. Machine learning (ML) and artificial intelligence (AI) techniques are increasingly being applied to analyze vast datasets and discover hidden patterns. Additionally, efforts to implement digital twins, that is, virtual replicas of physical materials or structures, facilitate predictive modeling and testing under different conditions.

These developments, especially with regard to data acquisition, processing, and management connected to facilitated reusage ("downstream usage"), are valid for and affect all (mechanical) test methods.[18] Hence, the considerations toward digitalized, machine-actionable, and semantically supported recording and processing of test values in this study may be universally applicable to various test methods.

2.2. Tensile Testing

Tensile testing is a fundamental and widely used technique in the field of MSE to evaluate the mechanical properties of materials. Its historical development spanning over decades, coupled with modern advancements, has led to its widespread use in research, industry, and quality control.[19–21] It involves subjecting a sample of the material to an axial load (tensile force), generally to fracture, while measuring the resulting deformation to determine one or more intrinsic mechanical properties.[5] This process provides valuable information on the behavior of a material under tension as well as important characteristic values used for the appropriate design of technical components, such as its strength, ductility, elasticity, and fracture properties.

Hence, tensile testing plays a crucial role in materials research for reasons of material characterization, performance prediction, material selection, quality control, as well as general material research, development, and design.[22–26] Standards on tensile testing, such as, for example, the tensile test of metals at room temperature in ISO 6892-1 (current version:[5]), define parameters such as loading rates, test piece dimensions, and information to report.

Further development of tensile testing comes along with developments in the field of data acquisition and management. Moreover, the historical evolution of data acquisition in tensile testing reflects the broader trajectory of technological advancements in MSE, transitioning from manual methods and handwritten notes to highly automated and sophisticated data acquisition and analysis processes. Starting from manual stress and strain calculations, early automation involved testing machines equipped with dial gauges and load cells[27] and the introduction of x–y plotters. With the advent of computers, data acquisition began to shift toward digital methods. Load and deformation measurements were recorded digitally, often stored in simple and various file formats. This transition marked a significant advancement in terms of accuracy and early data management. Connected thereto, the use of software tools enabled researchers to perform data analysis and downstream usage more efficiently. Load–deformation curves, stress–strain plots, and other mechanical properties could be generated and visualized digitally. However, data transfer between testing machines and computers often required manual input and was simply impossible or cumbersome, always leaving room for potential errors.[28]

The current advent and adoption of electronic lab notebooks (ELNs) and laboratory information management systems (LIMS) streamline the data generation and management process (see Section 2.3).[29–31] Researchers may now directly import data from testing machines into digital platforms, eliminating the need for manual transcription. This increases accuracy and improves traceability of data. As MSE steadily becomes more data intensive, the need to enhance their interoperability and machine understandability is apparent. SWT, including ontologies and linked data, are currently being increasingly employed to establish logical relationships between different types of data. This allows for more sophisticated analysis, correlation, and prediction on a computer-generated and machine basis intended to foster the work of researchers.

2.3. ELN and LIMS

To facilitate effective data management and knowledge integration, ontologies have emerged as valuable tools for organizing and representing information, especially with respect to characterization methods (see Section 1). Furthermore, the interconnection of ELNs and LIMS with ontologies offers a powerful approach to streamline experimental workflows and enhance data traceability even more. This also means that connecting the data acquisition using ELN and LIMS to the data management and further processing by applying an ontology leads to a fully integrated digital data pipeline ("virtual testing lab").[32] Therefore, consideration and integration of ontologies in ELNs and LIMS would be desirable and useful.

Since their introduction in the 1970s, ELN and LIMS have undergone significant advancements to become powerful tools for managing laboratory workflows and data.[18] While traditionally employed in clinical and pharmaceutical labs, they have nowadays found application in various scientific disciplines.[19] However, their direct adoption in MSE faces challenges due to the unique nature of materials research. To address this, specialized ELNs and LIMS platforms tailored to MSE are needed, incorporating advanced data analysis capabilities, integration with specialized instrumentation, flexibility for experimental design, and seamless collaboration functionalities. This can be provided by applying associated ontologies. As a result, modern ELNs have evolved to meet these needs, supporting real-time data transfer from advanced instruments, accommodating diverse data formats, and offering extensive customization options for researchers. Integration with modeling and simulation tools supported by facilitated data management due to ontology involvement further enhances the synergy between experimental and computational approaches in MSE. Hence, being the basis to build such LIMS pipelines, the creation of meaningful ontologies describing and representing materials characterization methods is crucial.

3. Semantic Web Technologies and Ontologies

Ontologies offer a systematic and structured approach to address data-related issues. In this context, data generated during tests as well as metadata, especially valuable concerning the reliability and reproducibility of experiments and results, are incorporated. Metadata provide important information on, for example, tools, machines, institutions, environmental conditions, and intentions involved in a test process. Thus, they are considered crucial in experiment description and may even lead to the possibility of uncovering unknown and unexpected effects and relations that were not in the main focus of an experiment.

3.1. Benefits of SWT and Ontologies for Test Data Management

The usage of ontologies and SWT when describing experimental data, such as those originating from materials characterization tests, especially leads to benefits concerning 1) data structuring, 2) reliability, 3) reproducibility, and 4) data reuse, integration, and knowledge preservation.

By formalizing domain knowledge and capturing relevant concepts, ontologies enable consistent data representation and support data structuring at a very granular level depending on the depth of modeling. Classes and properties are defined and thereby, a common understanding of processes and resulting data is established, which leads to facilitated data integration. A unique identification and elucidation of entities and relations is mandatory to meet the FAIR principles. As such, ontologies provide a common semantic framework for the description of MSE experiments and data.

By providing a common language for describing materials and experiments, ontologies promote the standardization of data. This standardization enhances data interoperability across different research groups, institutions, and databases. By adhering to established ontologies, researchers can seamlessly exchange data, compare results, and build upon existing knowledge. MSE research often involves collaborations across multiple disciplines for which a shared understanding and facilitated interdisciplinary communication is provided by structured and quasistandardized data. Researchers from diverse domains can collaborate, exchange data, and combine their expertise to gain deeper insights into materials properties and behavior. Furthermore, data discovery is facilitated by enabling semantic search and retrieval which is even possibly performed by machines. By incorporating semantic annotations using ontological concepts, experimental data becomes more easily discoverable and accessible. As a result, relevant datasets can be identified efficiently, data reuse is fostered, and scientific progress is accelerated. In addition, the integration of heterogeneous datasets is enabled, including experimental results, computational models, and literature data. By mapping data to ontological concepts (“data mapping”), diverse data sources can be unified which enables comprehensive analyses and cross-domain investigations. This integration enhances the efficiency of data-driven materials research such as in simulation processes or analyses that include AI approaches.

When modeled accordingly, ontologies provide the capacity of capturing and encoding detailed experimental metadata, including sample preparation methods, instrument settings, and measurement protocols. By explicitly documenting these parameters using agreed ontological terms, experiment reproducibility is enhanced as experiments based on the recorded metadata can be precisely replicated which ensures and preserves the accuracy and reliability of the results. Moreover, explicit guidelines and descriptions of the experimental process are provided since they can be regained from the ontology-based description. Such ontologically represented protocols can be followed to reduce ambiguity and enhance the likelihood of obtaining consistent results to validate scientific findings and promote transparency in materials research.

Finally, ontologies serve as valuable knowledge repositories, ensuring the preservation and dissemination of experimental findings. By capturing the semantics and context of data, they facilitate long-term data preservation in a simple and contextual form which enables future researchers to access, understand, and build upon past experiments. This promotes scientific progress and reduces redundancy in experimental efforts. SWT and the usage of ontologies in the field of MSE hold great potential for advancing materials research and accelerating scientific discovery.[33]

3.2. Resource Description Framework (RDF) and Triples

In the realm of SWT, the resource description framework (RDF)[34] created by the RDF working group[35] of the World Wide Web Consortium (W3C)[36] is a fundamental technology that enables the representation and exchange of knowledge in a structured and machine-understandable manner. It was developed as a standard model for data interchange on the web.[37–39] RDF extends the linking structure of the web to use uniform resource identifiers (URIs) to name the relationship between things as well as the two ends of the link which is usually referred to as a “triple” (see **Figure 1**). This simple model allows structured and semistructured data to be mixed, exposed, and shared across different applications.

This linking structure forms a directed graph in which the edges represent the named link between two resources that are represented by the graph nodes. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations.

Hence, RDF employs a simple and powerful data model in its core, organizing information as subject–predicate–object triples.

In this paradigm, each triple consists of the three components. 1) Subject: The subject represents the resource or entity being described denoted by URI to unambiguously identify it. It serves as the focal point of the triple and forms the foundation for

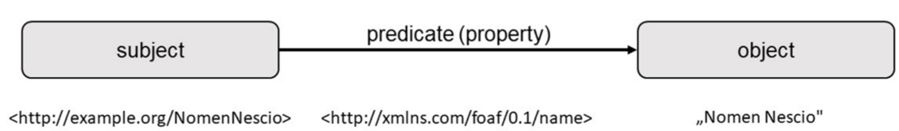

Figure 1. Triple of subject–predicate (property)–object with example according to RDF.

ADVANCED ENGINEERING MATERIALS
www.aem-journal.com

expressing relationships with other resources. 2) Predicate (Property): In the style of the syntax of natural languages, the predicate represents the relationship or attribute between the subject and the object. Predicates are also identified by URIs and define the semantics of the relationship. Therefore, in SWT, they are usually referred to as "object properties" if pointing to resources or blank nodes or "data properties" if pointing to literals. 3) Object: The object represents the "value" or a conceptual object of the property and can be either another resource identified by URI or a literal value, such as, for example, strings (concatenation of characters) or (scalar) numbers.

In the triple given exemplary in Figure 1, the subject is represented by the URI *http://example.org/NomenNescio*, referring to an individual named *Nomen Nescio*. The predicate in between them, denoted by the URI *http://xmlns.com/foaf/0.1/name*, indicates the property "name" from the Friend of a Friend (FOAF)[40] ontology vocabulary.

The object is the literal value *"Nomen Nescio"* which represents the name of the individual.

RDF triples form a graph-based data structure, allowing for the creation of interlinked and interconnected knowledge graphs. This graph-based representation fosters the flexibility and expressiveness of RDF, enabling the seamless integration of diverse datasets and the creation of a web of interconnected knowledge.

To represent RDF data in a human readable and writable format, the turtle (triple in turtle [TTL]) notation is often employed. TTL is a widely used serialization format for RDF that provides a concise and intuitive syntax for encoding RDF triples and constructs such as lists and blank nodes (BNodes).[41,42] In TTL, triples are expressed in a subject–predicate–object order, with each triple terminated by a period. Furthermore, it allows for the use of prefix declarations to abbreviate long URIs to improve readability.

An example of the RDF triple in Figure 1 in TTL notation is as follows (**Figure 2**).

Therein, the prefix declaration *@prefix foaf: http://xmlns.com/foaf/0.1/* allows to use *foaf:* as a shorthand for the FOAF namespace URI (Figure 2).

The triple itself is written similarly to the previous example, making it more concise and user friendly.

All in all, the subject–predicate–object triple model of the RDF serves as the foundation for representing knowledge on the Semantic Web. The turtle notation provides a convenient and efficient way to encode RDF triples while facilitating the creation and exchange of structured knowledge in a clear and concise manner. Hence, the Semantic Web fostering a new era of data integration, knowledge discovery, and enhanced machine understanding grounds its success on the usage of RDF.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
<http://example.org/NomenNescio> foaf:name "Nomen Nescio" .
```

Figure 2. Exemplary RDF TTL notation.

3.3. Types of Ontologies

In connection with the digital transformation to facilitate and increase the efficiency of materials development and materials design, the focus is particularly on the reuse or re-evaluation of data on materials and processes in different contexts. This requires interoperability of data in order to be able to use them in different application areas and tools. Therefore, and to comply with the FAIR principles for data management, the creation of references and relations of semantic concepts between different ontologies is expedient in addition to a data structuring that is based on standardized vocabulary as far as possible.

Accordingly, modularizable and extensible ontologies are crucial tools for implementing the FAIR principles. They allow to semantically structure and annotate raw data, processed data, and context data using a shared, consistent, and understandable vocabulary based on fundamental concepts.[43–45] In this context, there are different types and levels of ontologies. 1) Top-level ontologies are overarching, agnostic, and describe general concepts that are useful in many domains. 2) Midlevel or core ontologies represent abstract domain concepts at an intermediate level that allow the complex and expressive domain ontologies to be interconnected. One of such is the core ontology of the MaterialDigital platform (PMD Core Ontology [PMDco]).[46,47] 3) Domain or domain-specific ontologies are developed based on explicit expert knowledge and represent concepts that belong to specific domains, for example, distinct processes, research, and testing methods.

Thus, this knowledge is prepared in a generally understandable and sustainable way. With the help of a link between the created domain and higher-level ontologies, data from different sources can be easily found, shared, reused, and analyzed based on commonly used vocabulary. Furthermore, various alternative ontologies may exist for one and the same subject matter within a domain, such as test methods, production processes, and material descriptions. They may refer to each other extensively or partially, be linked to each other, or exist independently. These can be selected arbitrarily and deliberately by users if they are suitable with respect to the corresponding requirements. When developing ontologies describing standardized test procedures, such as the tensile test, consistency should always be sought, that is, all concepts, their relations, symbols, and units leading to correct syntax and semantics should be defined following the associated test standard, and references to the corresponding glossary should be given.

3.4. Terminology, Assertions, and Shapes

To exploit the full potential of data combined with semantic technologies, DL are to be used. DL is a family of knowledge representation formalisms used in various digitalization fields, such as AI and computer science, to model and reason about the concepts, relationships, and properties of objects within a domain. DLs are widely employed in SWT, ontology engineering, and knowledge-based systems.[48–52]

The two main components of a DL knowledge base are the terminological box (T-Box) and the assertional box (A-Box). The T-Box defines the terminology of a knowledge base. It is a set of terminological axioms that describe the hierarchical relationships, classes, and general concepts within the domain. The T-Box contains statements that assert subsumption relationships between classes by representing how one class is related to another. To express such relationships, object properties (see Section 3.2) are used. These relationships establish a taxonomy or a partial order among the classes connected by object properties. Typically, the T-Box includes the following types of axioms: 1) subsumption axioms specify that one class is a subclass of another; 2) equivalence axioms assert that two classes are equivalent, meaning they have the same instances; 3) disjointness axioms specify that two classes have no common instances; and 4) role hierarchy axioms define hierarchical relationships between roles (properties) and may be included in the T-Box of more expressive DLs. Hence, the T-Box provides the structural framework for organizing knowledge that enables automated reasoning to infer new facts based on the relationships defined in the taxonomy.

The A-Box represents the assertional knowledge about individuals (individual instances or objects) within the domain. It contains assertions about the membership of instances in different classes and the relationships between instances.

A-Box statements usually include 1) concept assertions indicating that a specific individual belongs to a particular class and 2) role assertions describing the relationships between individuals.

The A-Box allows the representation of specific facts and instances within the domain, enabling the knowledge base to describe concrete situations or instances of the concepts defined in the T-Box. As such, the T-Box defines the fundamental existence and possible occurrence of things that are represented in the A-Box, if there is a manifestation thereof that needs to be considered. For instance, a tangible tensile testing machine physically present in a laboratory could be digitally represented as *tte:tensileTestingMachine_1* in the A-Box as an instance of the (T-Box) concept (class) for the tensile testing machine *tto:TensileTestingMachine*. This A-Box instance *tte:tensileTestingMachine_1* could then be linked to other A-Box instances, for example, to an instance *tte:grips_1*, which is assigned to the *pmd:Grips* class defined in the T-Box and linked to the instance *tte:tensileTestingMachine_1* via the object property *pmd:component*. For further details, see **Figure 3** in Section 4.3.

By combining the T-Box and A-Box, DL provides a robust and formal framework for knowledge representation and reasoning, allowing systems to make logical inferences, perform consistency checks, and answer complex queries within a given domain. DLs are the foundation for ontology languages and are essential for building intelligent systems.

Hence, semantic rules may be implemented to RDF graphs enhancing the expressiveness and interoperability of linked data. Powerful approaches to achieve this are through the utilization of standards such as the Shapes Constraint Language (SHACL)[53] and the general purpose modeling language for linked data linkML.[54]

SHACL is an integral part of the W3C Semantic Web standards and provides a mechanism to define constraints on RDF

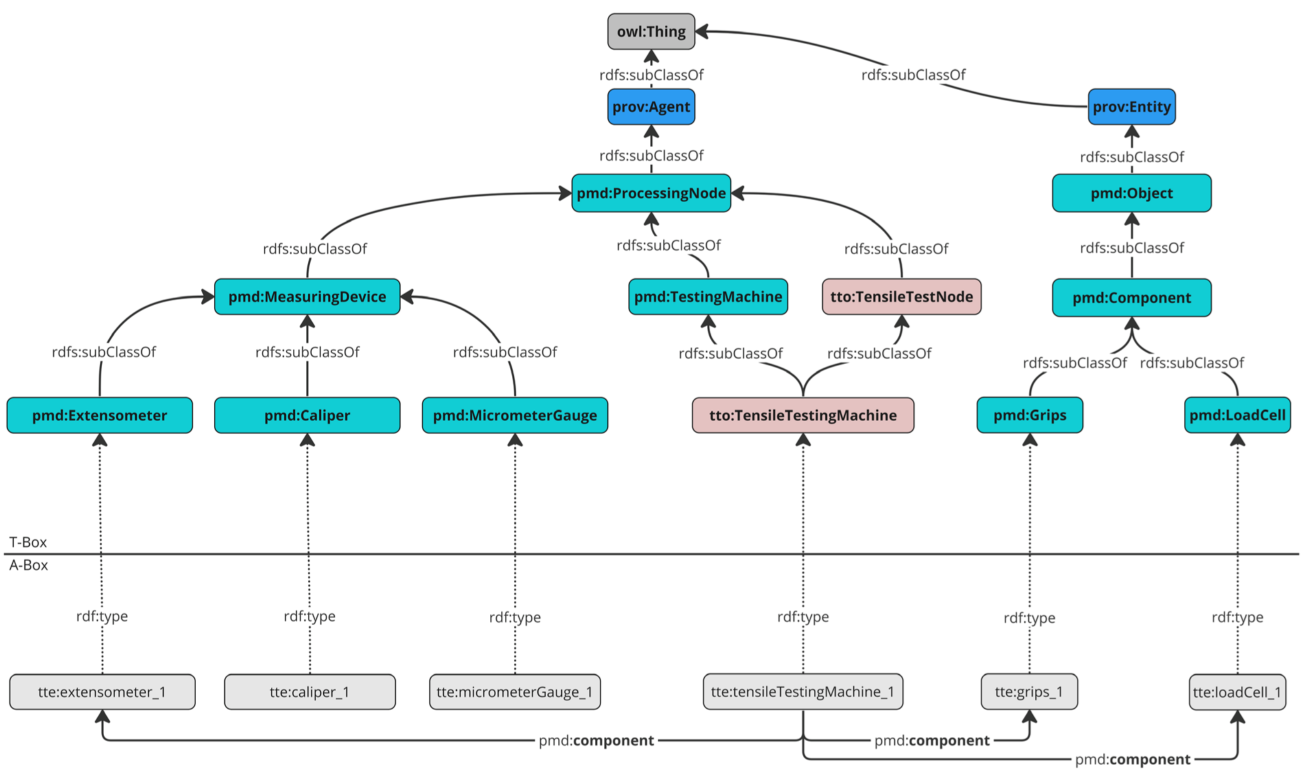

Figure 3. Processing nodes used in TTO; terminological box (T-Box) and assertional box (A-Box); specific color coding is used to differentiate ontology concepts/classes: gray: basic ontology class; blue: PROV-O classes; turquois: PMDco classes; salmon colored: TTO classes on T-Box level; light gray: instances on A-Box level.

graphs. These constraints, expressed as shapes, allow the specification of rules that A-Box instances have to adhere to, for example, governing the structure, datatypes, and cardinality of properties associated with specific classes. For instance, one can define that instances of a particular class should have a mandatory property or that a certain property must have values within a specified range. This enforces semantic integrity and facilitates more accurate and meaningful data representation. LinkML focuses the modeling and representation of linked data schemas. It provides a framework for defining ontologies and facilitates the incorporation of semantics into the modeling process. LinkML can be seamlessly integrated with SHACL to enhance the expressiveness of constraints and further refine the semantics associated with ontological classes. Accordingly, the combination of SHACL and linkML allows to create comprehensive and semantically rich ontologies and graphs.

3.5. Roadmap for Ontology Creation

When developing ontologies to semantically represent content matters such as test methods, production processes, and material descriptions, some general aspects may be considered, and a common path of ontology development (recommended guideline) can be followed.

In ontology development, sources of information and knowledge typically used in the fields of sciences and research are applied, such as scientific literature, standards describing materials and processes, technical manuals, and knowledge of (domain) experts. Representing the basis of research, scientific literature usually provides terms, denotations, definitions, and relationships of entities regarded in a specific domain. When developing ontologies for distinct procedures such as test methods, corresponding test standards are very helpful for the definition and description of parameters and entities within the ontology since they usually contain standardized definitions of the terminology and symbols used that are specific to the processes considered in a dedicated list within the standardization document. Furthermore, the relationships between entities, their explicit usage, and how they interact are described in the text body of the standard in natural language. In particular, the agreement made on terms and definitions by an expert standardization committee inherent to standards is beneficial. As another aspect, manuals of machines involved within processes may indicate distinct steps that need to be respected and thus, modeled in the ontology when describing those processes. Moreover, specific parameters, machine settings, and metadata helping to understand and reproduce the process may be obtained therefrom. Furthermore, the experience and knowledge of experts need to be considered during ontology creation likely allowing to represent requirements, actions, and specific information concerning processes in the best possible manner. Working with real-world data, systems, and applications, only domain experts possess the experience that provides valuable insights into the challenges, nuances, and complexities of representing domain knowledge in a formal ontology. By drawing on practical experience, experts can identify relevant use cases, scenarios, and requirements that shape the design and scope of the ontology. In particular, experts are able to identify patterns, structures, trends, and correlations underlying processes and resulting data to model. Hence, domain experts know every single action that is usually performed in the laboratory. Concerning tensile testing, for instance, the straight alignment of the specimens during clamping can be important, so that the introduction of a "grips alignment" process into the ontology may be advantageous since this process may affect the test results. Using this knowledge, clear, unambiguous concepts, and relationships can be defined within the ontology ensuring to capture the necessary information. Interdisciplinary knowledge and perspectives of domain experts may enrich the ontology by incorporating connections, analogies, and synergies.

Starting from these sources of knowledge, the process of ontology development in a specific domain can be performed. This includes the steps of 1) gathering information and identifying entities and parameters to be included, 2) structuring and finding categories and subcategories, 3) visualizing the process to be described for verification, 4) creating a thesaurus, and 5) converting this into the machine-actionable web ontology language (OWL) based on RDF.

The identification and listing of process-relevant parameters can be done using classical tabular tools and should include a categorization. In this respect, all necessary process parameters are supposed to be considered in detail, such as nominal parameters having necessarily to be set to run an experiment and process and measurements performed during the process. To add the correct syntax and semantics, symbols and units can be defined according to the standard, if applicable, and the definitions of terms as well as references to corresponding glossaries are to be given. For the subsequent creation and definition of a taxonomy as well as to find all required relations between concepts and entities, the use of a visualization is recommended. For this purpose, any drawing tools can be used, in principle, regardless of whether they are digital or analogous. However, digital tools include the benefits of facilitated sharing of documents and information between coworkers and simplified data handling in subsequent development steps. This is followed by the binding formalization in a thesaurus in which the definitions and relations of entities, concepts, and their relations are explicitly noted in natural language. Based on the thesaurus, representing controlled and shared vocabulary and knowledge within the specific domain regarded, all these considerations are finally transformed into an ontology using specific tools (Section 3.6). At the latest at this point, well-known resources should be checked for the availability of suitable ontologies that could be reused, for example, by browsing GitHub repositories[55] or ontology publication platforms such as MatPortal.[56] Moreover, and in any case, the semantic interconnection to higher-level ontologies, such as top-level and midlevel ontologies, should be considered. Due to the focus on SWT in recent years and the corresponding rapid development in this field, a lot of powerful ontologies were created that can be applied nowadays.[57,58] Following standardized requirements for top-level ontologies,[59] the basic formal ontology is even standardized in ISO/IEC 21838-2.[60] At least, a partial integration of semantic concepts well defined in such ontologies is worth considering. This way, further interoperability is facilitated (cf. Section 3.3).

3.6. Tools

3.6.1. Protégé

For ontology creation, the software tool "Protégé"[61] of Stanford University is recommendable. It is a widely used and powerful ontology development tool that allows the construction of a taxonomic order of concepts and properties in a hierarchical visualization. Foreign ontologies already existing can be easily imported directly or indirectly. Protégé is designed to support the creation, editing, and management of ontologies. It is advantageous when creating new classes, object properties, and data properties to the ontology as well as when including annotations to classes and properties such as labels, definitions, and information sources. Furthermore, if required, new knowledge can be derived with the help of reasoning and inference, that is, by creating implicit from explicit knowledge using DL. An intuitive and user-friendly interface makes Protégé accessible to both novice and expert ontology developers. Furthermore, it allows users to tailor the interface within the scope of functionalities to suit their specific needs and preferences. Another advantage of the tool is the support of a modular plug-in architecture, enabling the integration of additional features and extensions. Users can enhance the capabilities of the tool by installing various plug-ins for ontology visualization, reasoning, and more. In particular, automated reasoning is supported that allows to infer new knowledge based on the axioms and rules given in the ontology and, especially, helps in detecting inconsistencies and errors in the ontology with the "explain" feature, ensuring its overall quality.

3.6.2. Ontopanel

Among others, the graphical tool "Ontopanel"[62,63] can be used advantageously for visual ontology development. Ontopanel is a web-based plugin for the open chart drawing application diagrams.net (draw.io)[64] that simplifies the development of patterns, shapes, and schemata and method graph creation for domain experts. Originally designed within the frame of the Materials-open-Laboratory project[65] to address challenges faced with MSE domain ontology development, it is now versatile and applicable to various domains. The plugin consists of three tool parts: 1) "Library" for ontology conceptualization, 2) "EntityManager" for importing and reusing entities from existing ontologies, and 3) "Convertor" for converting method graphs into OWL and performing validation and data mapping. As such, Ontopanel combines the features of supplying users with a free drawing tool and allowing direct ontology conversion. The built-in converter transforms drawn ontology graphs directly into an RDF file including validation, that is, the convertor also checks for consistency with OWL rules. The Ontopanel service is available online, and its source code is openly shared on GitHub under the Apache-2.0 license. Hence, the key features of Ontopanel include easy graph creation, efficient data mapping, and error detection, streamlining the ontology development process and promoting broader acceptance of semantic data structuring for domain experts.

3.6.3. Scripts

In principle, ontologies can also be created by directly writing them triple by triple in a text-based format. This is not only possible but commonly done using RDF/XML, TTL, and Manchester syntax (cf. Section 3.2). Writing ontologies triple by triple can be especially useful for small-scale or simple ontologies, as it allows for direct and explicit control over the knowledge representation. Even small parts of larger ontologies may be directly written that way to create small examples for discussions in community-driven approaches facilitating the modeling of crucial knowledge. However, for large and complex ontologies, writing triples manually may become confusing, cumbersome, and prone to errors. For more extensive ontologies, a practical approach to generate the RDF triples programmatically is to use scripts, such as, for example, scripts based on Python programming language. Several RDF libraries are available, such as RDFLib, that facilitate working with RDF data and triples programmatically. Applying (Python) scripts, the ontology structure can be defined and classes, properties, individuals, and their relationships can be created easily using code. The script can then generate the corresponding RDF triples and output them to a text file or directly insert them into a triple store. Scripts can also be used for simultaneous data mapping. Therefore, data obtained in an arbitrary data format may be preprocessed, for example, by read-in and data transformation, and directly included in the graph, connecting concepts from the ontology and data. Hence, only one script may be needed to create RDF-formatted data connected to a specific ontology from test data that originated from several sources and obtained in different data formats. Such script-based data processing is particularly advantageous for original data in recurring structures, for example, when data from test methods are considered that are repeatedly performed in the same way in an institution.

Accordingly, data handling using scripts has several advantages such as automation, maintainability, reusability, and data integration. However, the validity and consistency of generated RDF triples with the intended ontology structure are essential to be ensured. Proper validation and testing are necessary and recommended to avoid errors in the resulting ontology ("ontology evaluation").

3.6.4. Ontology Evaluation

Various approaches are available to ensure data integrity and semantic coherence, especially addressing consistency with technical rules and domain-related data reliability, for example, after interconnecting data originating from different sources. For a technical, rule-based check, tools or libraries that validate RDF triples against the constraints specified in the ontology can be used, such as the OntOlogy Pitfall Scanner.[66] This tool automatically diagnoses OWL ontologies and helps developers to evaluate ontologies focusing on newcomers and those not familiar with DL and ontology implementation languages. Concerning data integrity, resulting files may be inspected manually. Such a human review of triples on a sample basis allows to identify inconsistencies that automated validation might miss. To support this, unit and sample data testing can be utilized.

Therein, various scenarios and edge cases can be covered, and datasets with known outcomes (RDF representations) may be used to ensure that the conversion logic produces valid RDF triples by querying and comparing the results with the expectations. Furthermore, data reconciliation techniques such as conflict resolution algorithms and semantic reconciliation can be used in some cases. The first aim to identify and resolve conflicts is by applying predefined rules, strategies, or optimization criteria but their selection depends on the nature of the data conflicts and the specific requirements of the application. The latter focuses on ensuring that concepts, properties, and relationships are aligned with the semantics given in the ontology. Thereby, the concept matching is supposed to be double checked, that is, the key maps and corresponding passages in the script connecting ontology concepts to those given in the data have to be checked for all different sources.

3.6.5. Data Transformation Pipeline

For the purpose of transforming data obtained from characterization methods to RDF data, the structure of a pipeline that consists of an interconnection of various tools for data handling is suitable. Therein, tools can be used sequentially, and such a pipeline may be advantageous when processing data that changes slightly in format, size, and number of files.

Such a pipeline was also created in the frame of the PMD project by developing and applying individual tools ("microservices"). Starting with data generated in characterization tests, all information belonging to a dataset is collected first and a native, disordered JavaScript Object Notation (JSON)[67] file is created. Again, the input files may originate from several data sources of arbitrary formats such as csv, xlsx, and pdf. JSON is a text-based and language-independent data exchange format. It was derived from the ECMAScript programming language standard and defines a small set of formatting rules for the portable representation of structured data. In particular, it is human and machine readable.

In order to organize terms and standardize the contents of the JSON file, it is converted into a canonical JSON (canonicalization) in a next step using the TTO. Therefrom, a key map was formed which helps to replace terms used in a given file with shared and distinctly defined terms (common vocabulary) for the purpose of harmonization and is thus specifically related to this file. Subsequently, the contents of the canonical JSON file can be linked to the semantic concepts by addressing URIs stored in an ontology. This mapping process links data values to their universally valid, uniform, and, if necessary, standardized name, to which a unique identifier is assigned (instance URI). This way, an RDF file is created which can be stored in a special database (triple store).

A significant advantage of building such a pipeline in the form of microservices (individual tools) is the possibility of using them from a certain arbitrary point or only certain microservices. For instance, if data from material characterization tests are already generated in a structured and unified form, possibly resulting from the use of an electronic lab book, a canonical JSON may be available, and an ontology mapping needs to be performed only. Furthermore, the triple store could be used only to publish semantically structured data that has already been generated using other methods.

4. Tensile Test Ontology (TTO) Development

Since standards provide a profound basis for the development of ontologies representing a distinct process by taking the classic steps within the ontology development path (Section 3.5), the well-known ISO 6892-1:2019-11[5] standard describing the tensile test of metals at room temperature was selected to develop the TTO. Furthermore, a "preferable data structure" in terms of the categorization of typical data resulting from a tensile test was developed in cooperation with the German standardization committee being directly involved in the development and revision of the corresponding test standard. This data structure depicts the view of MSE experts on how tensile test data is supposed to be categorized and which data may be recorded and provided by test operators to obtain a comprehensive tensile test dataset.

The TTO was designed on the basis of the midlevel ontology PMDco.[31,35] Accordingly, ontological PMDco concepts are reused and newly created concepts specific to the tensile test are connected thereto. This way, the TTO extends the PMDco. Vice versa, some general MSE and characterization method concepts originating from the consideration of the tensile test were directly included in the PMDco.

The PMDco is an ontology comprising a comprehensive vocabulary for MSE developed through community consensus and collaboration with MSE experts. It provides a standardized foundation for representing MSE concepts and knowledge in a structured manner, making it highly understandable for domain experts. The ontology incorporates midlevel classes that act as connectors between domain-specific and common higher-level ontologies (see Section 4.3, **Figure 4**). Its persistent and unique concept identifiers, enabling reliable and long-lasting referencing and linking of concepts, are accessible through the PMDco namespace *https://w3id.org/pmd/co/*. The class layout of the PMDco aligns with the higher-level PROV ontology (PROV-O) of the W3C which ensures a clear, well-organized, and reliable basis for the ontology. Furthermore, the PMDco leverages existing resources by incorporating elements from popular task and domain ontologies such as the ontologies of Quantities, Units, Dimensions, and Types (QUDT)[68] and Chemical Entities of Biological Interest.[69] This reuse of ontologies establishes a bridge between different knowledge domains and facilitates the representation and interoperability of tensile test data.

For the development of the TTO, Protégé, Ontopanel, as well as other visual tools mainly used for discussions within the MSE community were applied (Section 3.6).

4.1. Tensile Test Data Structure

Considering the ISO 6892-1 standard for the tensile test of metals at room temperature, a general data structure describing the tensile test was created. As a first level of categorization, tensile test data was divided into primary data, secondary data, and metadata. This first-level, basic data structure may also be universally

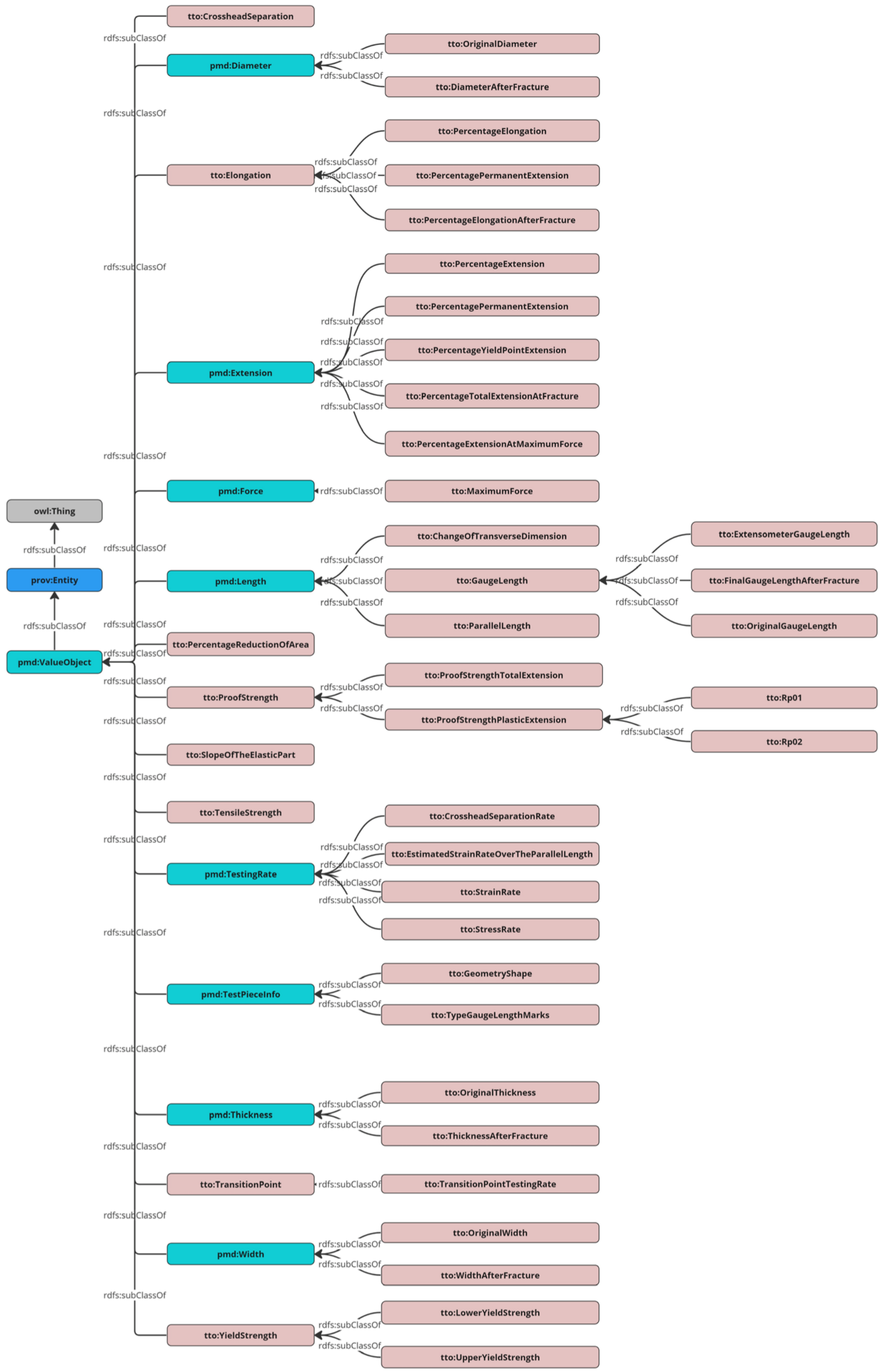

Figure 4. Value objects defined in TTO specific to the tensile test; specific color coding is used to differentiate ontology concepts/classes: gray: basic ontology class; blue: PROV-O classes; turquois: PMDco classes; salmon colored: TTO classes.

valid for other MSE characterization methods. Primary data was defined as data directly acquired during a tensile test by sensors such as, for example, time, force, and length, and completed by data acquired before and after a test referring to the geometry of the test piece such as, for example, the original and final thickness or diameter of a test piece, respectively. Hence, primary data describes data generated necessarily as a direct result of a test. Furthermore, primary data may also be denoted as "raw data."

Secondary data refers to characteristic values and results that were determined by equations and algorithms in a typical analysis procedure using primary data and metadata of a tensile test. Metadata describes attributes and additional data concerning the test such as the testing system, the test piece, and the laboratory which allow the evaluation of the quality and reliability of the measurements as well as a systematic search in a database. Following the usual definition of metadata being data that provides information about other data,[70] tensile test metadata is supposed to provide information on the specific tensile test performed and describe and relate the data obtained therefrom.

4.2. Fundamental Design Principles

In accordance with the general modeling approach of the PMDco, some fundamental design principles were followed in TTO development. As a result, TTO development follows a comprehensive approach that emphasizes interconnection, interoperability, and reusability. To achieve clarity, the classes and object properties are used in a condensed number to avoid unnecessary complexity, while utilizing generic and well-established concepts, classes, and object properties from higher-level ontologies such as the PMDco whenever possible. This also ensures consistency and compatibility with existing knowledge frameworks, including datatype definitions. For improved human readability, every concept in TTO is annotated by labels using *rdfs:label* (RDFS namespace: *http://www.w3.org/2000/01/rdf-schema#*) and definitions using *skos:definition* (SKOS namespace: *http://www.w3.org/2004/02/skos/core#*). If applicable, examples for the usage of the respective concept are incorporated via *skos:example*, providing practical instances for better understanding. Hence, class definitions are enriched with MSE domain knowledge, using both human-readable definitions in natural language (*skos:definition*) and implementing class relations. With respect to this, the reference to the standard is indicated in each case using the annotation property *pmd:definitionSource*. Human-readable identifiers, such as labels and class/object property designations, enhance queryability and make the ontology more accessible to users. By applying the PMDco as midlevel ontology, a solid foundation for the ontology is ensured and compatibility with other knowledge representations is fostered.

4.3. Classes and Properties

Following the general modeling approach of the PMDco and using the classes *pmd:Process*, *pmd:Object*, *pmd:ProcessingNode*, and *pmd:ValueObject*,[31] several classes were added that allow for a semantic representation of the tensile test. In particular, tensile test specific classes were included subordinate to the classes of "processing nodes" and "value objects" (Figure 3 and 4). Processing node types typically applied during a tensile test are depicted in Figure 3. Besides using classic measuring devices such as a caliper and a micrometer gauge to obtain the dimensions of a tensile test piece, a tensile testing machine is naturally essential to perform a tensile test. The tensile testing machine may feature additional components to be considered such as grips for test piece attachment, a load cell, and an extensometer. They can be semantically linked to the tensile testing machine using the object property *pmd:component*. As provided for in the PMDco, metadata information on the processing nodes such as serial numbers and other identifiers of the equipment, documents describing calibration status details, and working range as well as capacity information can be linked using the *pmd:characteristic* object property and the corresponding classes.

Being on the same semantic level as *pmd:TestingMachine*, the class *tto:TensileTestNode* was added as an additional intermediate class between *pmd:ProcessingNode* and *tto:TensileTestingMachine* to include another layer allowing an enhanced querying. This is in accordance with the modeling approach of explicitly describing processing nodes and assigning them to their intended usage. The class *tto:TensileTestNode* allows to query explicitly for the essential processing nodes required to perform a tensile test.

Most classes specific to tensile testing were added as subclasses of *pmd:ValueObject* (Figure 4). This is reasonable since the generic *pmd:ValueObject* class is intended to link processes, processing nodes, and associated input and output objects. As such, *pmd:ValueObject* instances are information carriers to specific data and metadata. Hence, the classes added with respect to TTO represent (meta)data required to describe the (characteristic) values, results, and additional test information usually obtained from tensile tests. Therefore, they are elementary components of the TTO. Some TTO classes specify already existing classes in the PMDco in more detail, such as those describing the dimensions of the test piece and the test setup, for example, *tto:OriginalDiameter*, *tto:DiameterAfterFracture* as subclasses of *pmd:Diameter* and *tto:GaugeLength*, and *tto:ParallelLength* as subclasses of *pmd:Length* with their own subclasses, respectively. Other classes were directly subordinated as subclasses to *pmd:ValueObject* such as, for example, *tto:TensileStrength* and *tto:YieldStrength*. However, the latter are related to the *pmd:Stress* class using semantic references to ensure consistency with MSE knowledge.

During the creation of the classes in the TTO, the data structure developed from the contemplation of the associated standard (Section 4.1) was advantageous, especially to represent the MSE knowledge in the correct way and to ensure completeness of information required.

When creating ontologies to describe characterization methods applied in MSE, specific considerations may arise, as in the case of the modulus of elasticity and the slope of the elastic part in a stress–strain curve. The modulus of elasticity is a general material property. Therefore, the class *pmd:ModulusOfElasticity* is included in the midlevel ontology PMDco. This property may be determined from a tensile test by analyzing the slope of the linear part of the stress–strain curve that represents the elastic deformation during the test. Hence, the class *tto:SlopeOfTheElasticPart* specific to the tensile test was introduced to TTO. Both classes are needed because only under certain conditions and following respective regulations given in the standard, the slope of the elastic part may be considered a value for the modulus of elasticity. In such a case, an individual (instance) created that describes such a value can be of the type of both classes (*tte:mE_Instance_1 rdf:type tto:SlopeOfTheElasticPart* and *tte:mE_Instance_1 rdf:type pmd:ModulusOfElasticity*). If requirements conforming to the standard for the determination of the modulus of elasticity from the slope of the elastic part in the stress–strain curve are not met, the individual describing the value may only be assigned to the *tto:SlopeOfTheElasticPart* class. Whether the requirements are met can be decided based on the metadata associated with the particular test setup being used in this case. This also shows the importance of metadata and their consideration within a semantic representation of an MSE characterization method such as the tensile test.

Besides several classes, the TTO defines the object property *tto:relatesToExtension* to extend the available set of object properties inherited from the PMDco (**Figure 5**). Since object properties in ontologies represent relationships between individuals (instances) of different classes, the *tto:relatesToExtension* object property is used to establish a specific relationship between the two classes *tto:ProofStrengthPlasticExtension* and *tto:PercentageExtension*. In this context, *tto:relatesToExtension* is an object property specific to the tensile test which allows for a more detailed description of the interplay between plastic deformation and elongation. More precisely, this object property supports in semantically representing the characteristic value of proof strength plastic extension such as, for example, $R_{p0.2}$.

A line-by-line exploration of the object property definition (Figure 5) provides insights into its application (the following object properties refer to the subject *tto:relatesToExtension*, respectively).

rdfs:subPropertyOf pmd:relatesTo:

The first line indicates that *tto:relatesToExtension* is a subproperty of the property *pmd:relatesTo* inherited from PMDco. This suggests that *tto:relatesToExtension* represents a more specific or specialized relationship compared to the more abstract *pmd:relatesTo* property.

rdfs:domain tto:ProofStrengthPlasticExtension:

This line specifies that the domain of the *tto:relatesToExtension* property is the class *tto:ProofStrengthPlasticExtension*, that is, this property is used to describe relationships between instances of *tto:ProofStrengthPlasticExtension* and instances of the target class.

rdfs:range pmd:PercentageExtension:

This line indicates that the range of the *tto:relatesToExtension* property is the class *pmd:PercentageExtension*. Thereby, the "target class" is defined. This means that the property is used to relate instances of *tto:ProofStrengthPlasticExtension* to instances of *pmd:PercentageExtension*.

rdfs:isDefinedBy <https://w3id.org/pmd/tto>:

This line points to the location where the property is defined, which is the ontology at https://w3id.org/pmd/tto (TTO) in this case.

rdfs:label "relates To Extension"@en:

This line provides a human-readable label for the property, which is "relates to extension" in English. This label provides a descriptive name for the property.

```
https://w3id.org/pmd/tto/relatesToExtension


@PREFIX tto: <https://w3id.org/pmd/tto/>
@PREFIX pmd: <https://w3id.org/pmd/co/>
tto:relatesToExtension rdf:type owl:ObjectProperty ;
            rdfs:subPropertyOf pmd:relatesTo ;
            rdfs:domain tto:ProofStrengthPlasticExtension ;
            rdfs:range pmd: PercentageExtension ;
            rdfs:isDefinedBy <https://w3id.org/pmd/tto> ;
            rdfs:label "relates to extension"@en .
```

Figure 5. Code fragment defining object property *tto:relatesToExtension* in OWL, turtle notation.

In general, material properties are modeled as OWL classes in the TTO. Hence, the class *tto:ProofStrengthPlasticExtension* represents a material property related to the proof strength of a material during a tensile test (relation between instances of property and test, respectively, implemented via *pmd:output*). The class *tto:PercentageExtension* represents a material property related to the extension of a material during a tensile test (relation between instances of property and test, respectively, also implemented via *pmd:output*). Extension, in this context, refers to the increase in length of a test piece when subjected to tensile forces. It is measured by applying an extensometer and expressed as a percentage of the extensometer gauge length (modeled as OWL class *tto:ExtensometerGaugeLength*). According to the test standard, the proof strength plastic extension must necessarily be related to the value of percentage extension which, in addition, must always be indicated. In particular, the well-known and often used value of $R_{p0.2}$ describes the proof strength plastic extension at an exact value of percentage extension of 0.2%. Hence, the individual describing the $R_{p0.2}$ value (instance of type *tto:Rp02*) has to be related to the OWL class *tto:PercentageExtension* and a value of 0.2%. To express this specific context, the *tto:relatesToExtension* object property can be used.

To achieve more expressivity, some classes were defined using the OWL "Equivalent To" clause. Some of these are given exemplary in **Table 1**.

4.4. Patterns and Shapes

Shapes and patterns are useful to unravel the intricacies of process and instance connectivity (see Section 3.4). Therefore, the PMDco used as higher-level ontology in this study offers essential patterns elucidating the relationships between processes and instances, intended to serve as a foundational aspect of RDF graph creation. In the TTO context, such patterns were used and adapted to the specific shapes encountered in tensile tests. This adaptation process is crucial in understanding and characterizing the connectivity between processes and instances within the unique context of tensile test shapes. The development of these specialized patterns for TTO adds a layer of specificity, addressing the nuances associated with the shapes involved in tensile testing procedures.

Parts of the manifestation of these patterns used in TTO-based graphs (A-Box) are depicted at a low granularity level in Figure 7. Hence, this visual representation offers a first glimpse on how these patterns are used when applied to the tensile test domain. Formal shape descriptions and constraints with the use of prevailing standards such as SHACL and linkML (see Section 3.4)

Table 1. Class expression axioms[78] included in TTO; the axioms are expressed in Manchester syntax with a short description given in natural language, respectively.

Class expression axioms	Description
Class *Tensile Test :participant* some *:TestPiece* and *:participant* some :TensileTestingMachine	The tensile test (process) has participants of type test piece and of type tensile testing machine; an instance with these participants is a tensile test.
Class *Rp0.2 :relatesToExtension* some (pmd:value value 0.2f)	$R_{p0.2}$ relates to an instance of percentage extension that has to have a value of exactly 0.2%.
Class *Elongation :relatesTo* some :OriginalGaugeLength	Elongation relates to the original gauge length, because the elongation is measured as a change of the original gauge length.
Class *Percentage Reduction of Area* ((*:relatesTo* some :OriginalThickness) and (*:relatesTo* some :OriginalWidth)) or (*:relatesTo* some :OriginalDiameter)	Percentage reduction of area relates to the cross-sectional area of the test piece which is defined by the test piece dimension such as diameter, thickness, and width, depending on the section shape of the test piece (circular, rectangular).

are not implemented so far. However, the patterns, initially derived from PMDco and partly adapted for TTO, are made available in a human-readable format using collaborative documentation tools. As such, the latter are designed as a work of reference for researchers.

5. Data Conversion: From the Lab to the End User

To make data available in an interoperable, machine actionable, and thus versatile manner, after their generation, they usually have to be transformed into appropriate data formats and stored in accessible databases. The transformation of data typically obtained from material characterization methods in various, arbitrary data formats into semantic data using ontologies is usually referred to as “data mapping” (cf. Section 3.1). More precisely, for each data individual, an instance is created that is supposed to be assigned to concepts provided by the ontology. Consistency with the ontology is ensured that way and the full potential of semantic interoperability can be used, having a universal data format available.

Hence, in general terms, the goal is to create structured, universally formatted, and searchable data for subsequent usage, such as evaluation (possibly in different contexts), simulation, and publication. The generated data usually originate from different sources, such as different material characterization methods, are populated by humans (e.g., manually entering single values) and machines (e.g., logged test series), and are mostly available in different data formats. These have to be unified and related to a common semantics for which ontologies are essential.

In this study, selected tensile test data (see Section 5.1) are mapped to the TTO (see Section 5.2) as a best practice example. Moreover, RDF data created therefrom were fueled in a triple store. Exemplary SPARQL queries were performed on this data (see Section 5.3) in order to show queryability and consistency.

5.1. Experimental Data

The experimental tensile test data used for data mapping in this study is part of a dataset provided in an open Zenodo repository.[71] This dataset comprises data obtained from a series of characterization tests performed on a plate of typical S355 (material number: 1.0577) structural steel (designation of steel according to DIN EN 10025-2:2019[72]). The tests include methods for the determination of several mechanical properties, one of which is the tensile test. The datasets were generated in the frame of the PMD project. Hence, this data is especially supposed to provide a basis for experimental data inclusion, conversion, and structuring (data management and processing) that leads to semantical expressivity, making it suitable for consideration in this study.

A total number of ten tensile tests including their analyses were performed in accordance with ISO 6892-1:2019[5] at the Bundesanstalt für Materialforschung und -prüfung, Berlin, Germany.

The dataset contains various types of data related to tensile tests. In accordance with the data structure and first-level ontology structure, the data are organized in the three main categories primary data, secondary data, and metadata. As primary data, tensile test measurements originating directly from the tensile test machine (“raw data”) comprising typical test series of “time”, “crosshead travel”, “force”, and “extensometer elongation” values including unit information are given in comma-separated value (CSV) format. The “secondary data” category includes diagrams generated by plotting the primary data, providing an overview of the behavior of each test piece and more detailed insights, for example, for determining specific characteristic values such as yield strength. Furthermore, test protocols and analysis results of all the tensile tests performed are contained which are given in XLSX data format. Metadata is given in the form of additional information on the test and the equipment used as a list and description of testing and measuring instruments stored in a PDF file. Hence, representing usual circumstances in nowadays material characterization efforts using classic test methods, a bunch of different information given in different data formats needs to be processed.

5.2. Data Mapping

In this study, an exemplary data mapping was performed using a dataset comprising information on a total number of ten tensile tests (cf. Section 5.1). For this data mapping, a Python script was used to write the data triple by triple (cf. Section 3.6). Within the script, the library RDFlib[73] well known to be advantageous for data mapping was applied. For the implementation and execution of the Python script, the environment Jupyter Notebook was used.

First, tensile test data was loaded from several source files in various data formats (csv, xlsx, pdf) using typical read-in techniques of pandas library. Then, the resulting tabular data frame containing all tensile test information available was iterated over and variables were created describing the characteristic values given in the data frame, respectively. Hence, each variable comprised one characteristic value or information including the respective unit, if applicable, (e.g., the value "6" and the unit "mm") that needed to be assigned to an instance describing a specific quality of the tensile test (e.g., "original thickness"). Having all variables, that is, all necessary information on a specific tensile test, available, instances were created by generating a URI and adding these instances to the knowledge graph, respectively. Thereby, the instances were assigned to one or more classes (using *rdf:type* object property) in accordance with the TTO, and values (and units) were assigned to the instances using the corresponding variables. Furthermore, just-created instances were interlinked with each other. For example, information on the dimensions of the tensile test piece was linked to the tensile test piece instance. Likewise, information on tensile strength, maximum force, and yield strength was linked as outputs to the tensile test process. Additionally, metadata such as information on the project the tests were performed for, the institute they were performed at, and the test machine they were performed with were linked to the tensile test process using the *pmd:characteristic* object property.

In **Figure 6**, an exemplary part of code is depicted that was used for adding the information on the tensile strength R_m obtained from one tensile test performed to the graph ("g") by defining various relationships and properties between different entities and values.

In the first line, a new variable of type URI reference named "RmIRI" is created. It concatenates the value of the experimentIRI, another URI previously created to describe the entire process (experiment), with the string "_tensileStrength" to form the new URI.

In the next line, the first triple is added to the graph "g." The triple consists of the subject "RmIRI," the *rdf:type* object property to assign instances (subjects) to a certain class in the ontology,

```
RmIRI = URIRef(experimentIRI + "_tensileStrength")
g.add( (RmIRI, rdf:type, pmd.TensileStrength) )
g.add( (RmIRI, rdf:type, pmd.SecondaryData) )
g.add( (RmIRI, rdf:type, pmd.Measurement) )
g.add( (RmIRI, pmd.value, Literal(Rm, datatype=XSD.float)) )
g.add( (RmIRI, pmd.unit, qudt.MegaPa) )
g.add( (testpieceIRI, pmd.characteristic, RmIRI) )
g.add( (processIRI, pmd.output, RmIRI) )
```

Figure 6. Part of code in Python script used for adding tensile strength (R_m) data to the graph "g"; within the inner brackets of the "add" method, a Python tuple is created from the three variables given, respectively.

and the object "pmd.TensileStrength." This way, it is stated that RmIRI is of type *pmd:TensileStrength* which is a class in TTO. The next lines state that the instance of "RmIRI" is also of type *pmd:SecondaryData* and *pmd:Measurement*. Using these types of assignments, an instance representing a characteristic value resulting from a characterization method is semantically described comprehensively in accordance with the PMDco modeling approaches. This way, the tensile strength instance is attributed to its MSE purpose by assigning it to the *pmd:TensileStrength* class which is further described in TTO as well as to its data and value scopes, respectively, to describe its function within the test procedure and add another level of queryability. At this point it becomes clear that an instance ("RmIRI" describing the characteristic value R_m) can belong to several OWL classes.

The line "g.add((RmIRI, pmd.value, Literal(Rm, datatype=XSD.float)))" adds a triple that associates the value in the variable "R_m," which is a floating-point numerical value, as a "literal" with the "RmIRI' URI. The property "pmd.value" (datatype property *pmd:value*) is used to denote the relationship between the resource and the value. Furthermore, a triple is added that links the "RmIRI" resource to the unit "MegaPa" (*qudt:MegaPa*, megapascals) in the subsequent line which represents the physical unit for the tensile strength (R_m) value.

In the last two lines, triples are added to the graph that link instances already created beforehand within the script in order to represent the tensile test process and the corresponding test piece to the instance created to describe the tensile strength. Accordingly, the "testpieceIRI" URI representing the tensile test piece instance is linked to the "RmIRI" URI applying the *pmd:characteristic* object property. Analogously, the "processIRI" URI is linked to the "RmIRI" URI making use of the *pmd:output* object property. In accordance with TTO and PMDco modeling, a characteristic value obtained from a characterization method such as the tensile test is semantically connected to the test process as being an output of the process and defined as a characteristic of the object (test piece) involved. This is reasonable, as the value is a direct result of the test process and would not have been known if the process had not been performed and since the value represents a description of a property inherent to the tested object which represents a specific material.

The result of this part of data mapping in the knowledge graph is sketched in **Figure 7**. Therein, ontology classes of PMDco, TTO, CSVW,[74] and QUDT are included in the T-Box as well as associated instances in the A-Box.

For the tensile strength (R_m) represented by the instance *tte:Rm_1*, a floating-point numerical value of 510 is given. The instance is furthermore linked to the instance *qudt:MegaPa* representing the unit megapascals in accordance with the QUDT ontology.

Moreover, the dataset containing information on the tensile test process (instance *tte:tensileTestProcess_1*) is instantiated in *tte:rawData_1* and linked as output to the process. The *tte:rawData_1* instance is further described by having a resource (*pmd:resource*) *tte:rawData_1_table* which is of type (*rdf:type*) *csvw:table*. The latter refers to the CSVW ontology used to describe tabular data.

The depiction of the semantic modeling in Figure 7 is representative for all characteristic values and information obtained from the tensile tests performed. The result of the data

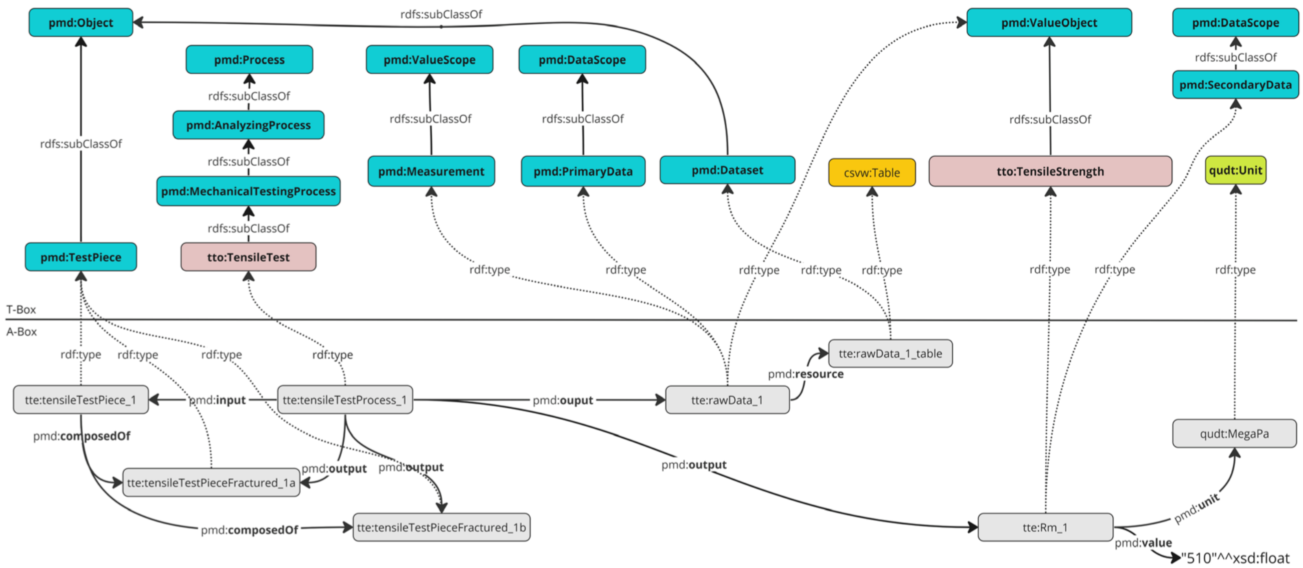

Figure 7. Semantic representation of the characteristic value of tensile strength (R_m) determined in a tensile test using TTO; specific color coding is used to differentiate ontology concepts/classes: turquois: PMDco classes, orange: CSVW classes, salmon colored: TTO classes, green: QUDT classes on T-Box level; light gray: instances on A-Box level.

transformation implemented in the script is an RDF graph. The graph can be queried using the semantic query language SPARQL.

5.3. SPARQL Queries

Having tensile test data transformed into RDF data and ready in a triple store, it can be queried using the semantic query language SPARQL (SPARQL Protocol and RDF Query Language). This was made a standard by the RDF Data Access Working Group[75] of the W3C and is recognized as one of the key technologies of the Semantic Web. SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns.

It enables unique addressing of each concept contained and the associated data values. SPARQL queries can be used directly in the web surface of a triple store, if applicable, or in scripts via an application programming interface (API) by addressing a SPARQL endpoint. In this study, both variants were applied. The Ontodocker[76] tool developed within the frame of the PMD project was used as triple store. It provides both, a web interface accessible by a simple web browser and an API for integration in a script. The SPARQL queries to be used are the same in both cases.

As an example (**Figure 8**), a SPARQL query was selected that is used to search for entities of type *pmd:TensileTest* (tensile test processes) that have a relationship with entities of type *pmd:TestPiece*. It is supposed to retrieve the corresponding *pmd:TensileStrength* entities, along with their associated *pmd:value* (bound in the variable ?rmVal) and *pmd:unit* (bound in the variable ?unit), respectively.

Accordingly, in the first line of the SPARQL query, the prefix "pmd" is defined and associated with the namespace URI "https://w3id.org/pmd/co/." This allows the shorthand notation "pmd:" to be used later in the query. In the SELECT statement, the variables are specified that the query will return as results.

```
PREFIX pmd: https://w3id.org/pmd/co/
PREFIX tto: <https://w3id.org/pmd/tto/>
SELECT distinct ?p ?s ?rmVal ?unit
WHERE {
   ?s a pmd:TestPiece .
   ?p a tto:TensileTest .
   ?p pmd:input ?s .
   ?p pmd:output ?output .
   ?output a tto:TensileStrength .
   ?output pmd:value ?rmVal .
   ?output pmd:unit ?unit .
} ORDER BY ?rmVal
```

Figure 8. Code fragment of SPARQL query to get a tensile test process (?p), the associated test piece (?s), the value of tensile strength (?rmVal), and the corresponding unit (?unit) in a table.

In this case, four variables are requested for which are ?p, ?s, ?rmVal, and ?unit. These variables are specified in the next part starting with the WHERE statement. Such a statement usually marks the start of the pattern matching section of a query in which the actual conditions for data retrieval are specified.

First, a pattern is defined where the variable ?s (subject of the triple) is to be of type *pmd:TestPiece*. Hence, triples are sought for which the subject is of type *pmd:TestPiece* which defines the first constraint leading to a filtering of results. Next, the variable ?p is supposed to be of type *pmd:TensileTest* and it has to have a relation *pmd:input* to the entity represented by ?s (next code line). With these first lines in the WHERE statement, the data in the triple store is filtered for test pieces that are connected to tensile tests which declare that information on tensile tests is

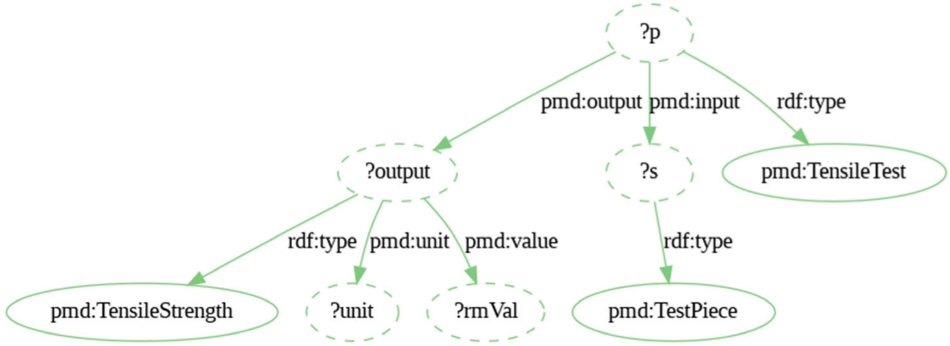

Figure 9. Visualization of the SPARQL query to get a tensile test process (?p), the associated test piece (?s), the value of tensile strength (?rmVal), and the corresponding unit (?unit) in a table; classes are outlined with a solid line, instances are outlined with a dashed line; visualization obtained directly from the Python script using sparqlgraphviz library.[77]

sought. In the next lines, the output of the tensile test that is to be found is denominated and specified. Therefore, a pattern is defined in which the variable ?p has a relation *pmd:output* to an entity represented by ?output. The latter is then required to be of type *pmd:TensileStrength*. For a human interpreter, it may become clear at this stage that tensile strength values obtained from tensile tests are supposed to be found with the query. To get the values connected to the instance (?output) of type *pmd:TensileStrength*, the next line defines a pattern where the variable ?output has a *pmd:value* property linking it to the value represented by ?rmVal. Thus, triples for which the subject ?output has a *pmd:value* property are sought. Finally, ?output is defined to have a *pmd:unit* property linking it to another variable represented by ?unit. After the WHERE clause, that is, the end of the pattern matching section, the "ORDER BY ?rmVal" line indicates that the query results should be ordered based on the values of the variable ?rmVal (numerical value of the *pmd:TensileStrength* entities, respectively), in ascending order.

A visualization of the SPARQL query (**Figure 9**) allows to trace the individual search steps and the connections, that is, the semantic references.

Table 2. Result of exemplary-selected SPARQL query to obtain R_m values and corresponding units associated with tensile test processes (experiments) and test pieces; namespace prefixes used are **tte:**https://w3id.org/pmd/resource/tto/ and **qudt:**http://qudt.org/vocab/unit/.

Process	Test piece	R_m value	Unit
tte:tto-tt-S355-1_process	tte:testpiece/Zx1	514	qudt:MegaPa
tte:tto-tt-S355-2_process	tte:testpiece/Zx2	504	qudt:MegaPa
tte:tto-tt-S355-3_process	tte:testpiece/Zx3	504	qudt:MegaPa
tte:tto-tt-S355-4_process	tte:testpiece/Zx4	519	qudt:MegaPa
tte:tto-tt-S355-5_process	tte:testpiece/Zy1	508	qudt:MegaPa
tte:tto-tt-S355-6_process	tte:testpiece/Zy2	507	qudt:MegaPa
tte:tto-tt-S355-7_process	tte:testpiece/Zy3	505	qudt:MegaPa
tte:tto-tt-S355-8_process	tte:testpiece/Zy4	514	qudt:MegaPa
tte:tto-tt-S355-9_process	tte:testpiece/Zd2	515	qudt:MegaPa
tte:tto-tt-S355-10_process	tte:testpiece/Zd3	511	qudt:MegaPa

Furthermore, the tabular result of the SPARQL query depicted in Figure 8 and 9 performed to the exemplary selected experimental data of this study (Section 5.1) is given in **Table 2**. For enhanced human readability, prefixes tte (**tte:**https://w3id.org/pmd/resource/tto/) as namespace for the associated instances and qudt (**qudt:**http://qudt.org/vocab/unit/) as namespace for corresponding units were defined to illustrate the results in Table 2.

It is worth mentioning that the processes, test pieces, and units are instances that can be semantically referred to, as can be seen from the usage of the respective URI (namespace prefix). Only the associated values of the tensile strength (R_m values) are given as literals (numerical values, "floats").

6. Conclusion

The digital transformation in MSE has led to significant advancements in materials development, design, and optimization, driven by computer simulations, AI, and ML. These technologies, along with continuous improvements in hardware and software, accelerate materials research. However, this digital paradigm shift poses challenges related to quality, interoperability, data reproducibility, and management. Furthermore, challenges arise in terms of user expertise, highlighting the need for user-friendly tools.

To address these challenges, SWT, including ontologies, have emerged as powerful tools in MSE. Ontologies provide a structured representation of domain-specific concepts and relationships, facilitating machine-actionable knowledge representations, data integration, and harmonization. In materials characterization, first efforts center on storing data according to standard-compliant semantic representations. Accordingly, associated ontologies are used to create interconnected knowledge graphs.

This study focuses on the TTO created within the PMD project, which semantically represents the method of tensile testing on metals at room temperature. The TTO is based on the PMDco and developed in accordance with the test standard ISO 6892-1. Hence, it provides a standardized vocabulary for tensile test data, ensuring interoperability, transparency, and reproducibility. Being developed with a focus on interconnection and reusability, this ontology includes classes and properties for a comprehensive representation of tensile test data structured in primary data, secondary data, and metadata. The interoperable data created using TTO enables tracking and tracing of experimental parameters and results which additionally ensure scientific integrity and support the verification of findings. The structured layer of TTO further ensures unified data storage, supporting retrieval and downstream usage of the latter. An integration with PMDco enhances interoperability across other domains by facilitating connectivity to these domain-specific ontologies, knowledge, and data. Persistent unique identifiers assigned for the TTO concepts ensure reliable referencing and linking over extended periods of time to ensure the continuity of scientific knowledge on tensile testing.

By introducing TTO and documenting its development process within this study, a best practice example on an effective and straightforward approach to creating an ontology is provided to the MSE community. Exemplified by the semantic representation of the tensile test on metals at room temperature, basic ontology development steps involving identifying parameters, structuring, visualization, thesaurus creation, and formalization are described in detail. Development tools such as Protégé, Ontopanel, and Python scripts simplifying ontology creation and data mapping are introduced, as well. These guidelines are supposed to support the MSE community in developing ontologies for their own experiments and processes. In addition to the TTO, tensile test data[71] and technical documentation are also publicly available for direct reuse by the MSE community.

Thinking ahead, SWT is becoming increasingly important in MSE due to their benefits for data management, knowledge representation, and collaboration. Data management based on semantic representations such as the TTO allows researchers to access, understand, and build upon past experiments which foster scientific progress. Future developments may involve integrating ELNs and LIMS with ontologies to create seamless data pipelines.

Acknowledgements

Valuable discussions on the structuring of data involving participants of the Innovation-Platform MaterialDigital (PMD; materialdigital.de) as well as of the ad-hoc working group on digitalization of the DIN standardization committee “NA 062-01-42 AA” are especially acknowledged.

Open Access funding enabled and organized by Projekt DEAL.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization was done by M.S., P.H., P.D.P., and B.S. Writing the manuscript was done by M.S., B.S., and P.D.P. Visualization was done by M.S. Ontology creation and data structuring were done by M.S., B.B., H.B., J.W., P.H., and P.D.P. Writing the review and editing was done by all authors.

Data Availability Statement

The data that support the findings of this study are openly available in Full dataset of several mechanical tests on an S355 steel sheet as reference data for digital representations at https://doi.org/10.5281/zenodo.6778335, reference number 6778336.

Keywords

data interoperability, domain ontology development, finable, accessible, interoperable, and reusable data management, knowledge representations, semantic web technologies, tensile testing

Received: January 19, 2024
Revised: March 20, 2024
Published online:

[1] M. Alam, F. Dittmann, M. Niebel, J. S. Lehmann, D. Dessí, J. F. Morgado, P. V. Hartrott, C. Eberl, P. Gumbsch, H. Sack, *Mater. Sci.* **2020**, https://api.semanticscholar.org/CorpusID:220902452.
[2] J. Kimmig, S. Zechel, U. S. Schubert, *Adv. Mat.* **2021**, *33*, 2004940.
[3] L. Lin, P. S. Kollipara, Y. Zheng, *Mater. Today* **2019**, *28*, 49.
[4] M. Wilkinson, M. Dumontier, I. Aalbersberg, J. Jsbrand, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, *Sci. Data* **2016**, *3*, 1.
[5] Materials Genome Initiative, Materials Genome Initiative, **2014**, https://www.mgi.gov/ (accessed: 03 March 2024).
[6] National Institute for Materials Science, DICE Materials Data Platform with Greatly Improved Data Collection and Accumulation Capabilities, **2023**, https://www.nims.go.jp/eng/news/press/2023/01/202301170.html (accessed: 03 March 2024).
[7] Centre national de la recherche scientifique, The 'Diadem' (materials) exploratory PEPR, **2023**, https://www.cnrs.fr/en/pepr/pepr-exploratoire-diademe-materiaux (accessed: 03 March 2024).
[8] Fundación IMDEA Materiales, materplat, **2023**, https://materplat.org/ (accessed: 03 March 2024).
[9] Materials Commons, Materials Commons 2.0, **2020**, https://materialscommons.org/ (accessed: 03 March 2024).
[10] re3data.org, Registry of Research Data Repositories, **2020**, https://doi.org/10.17616/R3D (accessed: 03 March 2024).
[11] U.S. National Science Foundation (NSF), NSF invests $26.7 million in building the first-ever prototype open knowledge network, **2023**, https://new.nsf.gov/tip/updates/nsf-invests-first-ever-prototype-open-knowledge-network (accessed: 03 March 2024).
[12] EMMC ASBL European Materials Modelling Council, The European Materials Modelling Council, **2014**, https://emmc.eu/ (accessed: 03 March 2024).
[13] NFDI Consortium, NFDI Nationale Forschungsdaten Infrastruktur, **2020**, https://www.nfdi.de/?lang=en, (accessed: 03 March 2024).
[14] NFDI-MatWerk Consortium, NFDI-MatWerk, **2021**, https://nfdi-matwerk.de/ (accessed: 03 March 2024).
[15] Platform MaterialDigital, Material Digital - The Material Digitalization Platform, **2019**, https://materialdigital.de/ (accessed: 03 March 2024).
[16] ISO 6892-1:2019-11, Metallic materials — Tensile testing — Part 1: Method of test at room temperature, International Organization for Standardization (ISO), Geneva, Switzerland **2019**.
[17] T. Austin, C. Bullough, D. Gagliardi, M. Loveday, D. Leal, A Guide to the Development and Use of Standard Compliant Data Format for Engineering Materials Test Data, **2010**, Retrieved from https://www.researchgate.net/publication/262047187_A_Guide_to_the_Development_and_Use_of_Standard_Compliant_Data_Format_for_Engineering_Materials_Test_Data, (accessed: 15 December 2023).

[18] M. Head, in *1998 IEEE AUTOTESTCON Proc. at IEEE Systems Readiness Technology Conf. Test Technology for the 21st Century (Cat. No.98CH36179)*, **1998**, pp. 479–485.
[19] T. Young, *A Course Lectures on Natural Philosophy and the Mechanical Arts*, Taylor and Walton, London, England **1807**.
[20] Martens, *Handbuch Der Materialienkunde Für Den Maschinenbau*, Julius Springer, Berlin **1898**.
[21] K. Hentschel, *NTM Zeitschrift für Geschichte der Wissenschaften, Technik und Medizin* **2011**, *19*, 5.
[22] P. Zheng, R. Chen, H. Liu, J. Chen, Z. Zhang, X. Liu, Y. Shen, *Fusion Eng. Des.* **2020**, *161*, 112006.
[23] G. E. Lucas, G. R. Odette, M. Sokolov, P. Spätig, T. Yamamoto, P. Jung, *J. Nucl. Mater.* **2002**, *307–311*, 1600.
[24] V. Karthik, K. V. Kasiviswanathan, P. Visweswaran, A. Vijayaragavan, K. Laha, T. Jayakumar, B. Raj, *Nucl. Eng. Des.* **2018**, *330*, 538.
[25] M. Rund, R. Procházka, P. Konopík, J. Džugan, H. Folgar, *Procedia Eng.* **2015**, *114*, 410.
[26] Introduction to Tensile Testing (Ed: J.R. Davis) *Tensile Testing*, ASM International, Michigan **2004**, https://doi.org/10.31399/asm.tb.tt2.9781627083553.
[27] J. Harding, E. O. Wood, J. D. Campbell, *J. Mech. Eng. Sci.* **1960**, *2*, 88.
[28] D. H. Carter, W. S. Gibbs, in *Symp. on Automation of Mechanical Testing, Amer Soc Testing and Materials*, Pittsburgh, PA, **1992**, pp. 28–39.
[29] R. R. Mahaffey, in *LIMS: Applied Information Technology for the Laboratory* (Ed. R.R. Mahaffey), Springer US, Boston, MA, **1990**, pp. 207–223, https://doi.org/10.1007/978-1-4684-9105-0_16.
[30] H. C. Faulkner, R. H. Farmen, R. S. Myer, E. M. Hahn, J. A. Rouse, *Chemom. Intell. Lab. Syst.* **1991**, *13*, 211, .
[31] H. K. Machina, D. J. Wild, *Jala* **2013**, *18*, 264, .
[32] H. B. Nasrabadi, T. Hanke, M. Weber, M. Eisenbart, F. Bauer, R. Meissner, G. Dziwis, L. Tikana, Y. Chen, B. Skrotzki, *Comput. Indust.* **2023**, *153*, 104016, .
[33] A. Valdestilhas, B. Bayerlein, B. Moreno Torres, G. A. J. Zia, T. Muth, *Adv. Intell. Syst.* **2023**, *5*, 2300051, .
[34] RDF Working Group, Resource Description Framework (RDF), **2014**, https://www.w3.org/RDF/ (accessed: 16 January 2024).
[35] W3C, RDFCore working group, **2003**, https://www.w3.org/2001/sw/RDFCore/ (accessed: 16 January 2024).
[36] W3C, W3C Semantic Web Activity, **2013**, https://www.w3.org/2001/sw/#rdf/ (accessed: 16 January 2024).
[37] C. Bizer, T. Heath, T. Berners-Lee, *Int. J. Semant. Web. Inf.* **2009**, *5*, 1.
[38] S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau Jr, S. Auer, J. Sequeda, A. Ezzat, *W3C RDB2RDF Incubator Group Rep.* **2009**, *1*, 113, https://www.researchgate.net/publication/240615202_A_Survey_of_Current_Approaches_for_Mapping_of_Relational_Databases_to_RDF.
[39] S. Sakr, G. Al-Naymat, *SIGMOD Rec.* **2010**, *38*, 23.
[40] D. Brickley, L. Miller, FOAF Vocabulary Specification 0.99, **2014**, http://xmlns.com/foaf/0.1/ (accessed: 03 March 2024).
[41] V. Carchiolo, M. Malgeri, G. Mangioni, *J. Comput. Syst. Sci.* **2003**, *66*, 293.
[42] L. Apvrille, J. P. Courtiat, C. Lohr, P. D. Saqui-Sannes, *IEEE Trans. Software Eng.* **2004**, *30*, 473.
[43] B. Bayerlein, T. Hanke, T. Muth, J. Riedel, M. Schilling, C. Schweizer, B. Skrotzki, A. Todor, B. Moreno Torres, J. F. Unger, C. Völker, J. Olbricht, *Adv. Eng. Mater.* **2022**, *24*, 2101176.
[44] R. Arp, B. Smith, A. Spear, *Building Ontologies with Basic Formal Ontology*, The MIT Press, https://www.google.com/search?sca_esv=83a32caac141be1d&rlz=1C1GCEU_enIN1078IN1078&q=Cambridge,+Massachusetts&si=AKbGX_paaCugDdYkuX2heTJMr0_FGRox2AzKVmiTg2eQr2d-rsuO9-E1IsiKWdB37jCF-c1baWZ2u4VJkVg1LgNDCqaj-pH-fRVfvmhgnfKg5hGUwXuConTbum-ZmWGoGlVbGvP-cIF-jJDECUE3BB7jauVMiSIycgnGvZduC5uAPeOkd-kGYxM4TkjoqATtKLQz8-P6SA_tmhBKQlTTjx-2pBboFadufQ%3D%3D&sa=X&ved=2ahUKEwiwpee23I-FAxXUamwGHVNoCQ0QmxMoAXoECFYQAw Cambridge **2015**, ISBN: 9780262527811.
[45] C. Partridge, A. Mitchell, A. Cook, J. Sullivan, M. West, *A Survey of Top-Level Ontologies - To Inform the Ontological Choices for A Foundation Data Model*, University of Cambridge, https://www.google.com/search?sca_esv=83a32caac141be1d&rlz=1C1GCEU_enIN1078IN1078&q=Cambridge,+Massachusetts&si=AKbGX_paaCugDdYkuX2heTJMr0_FGRox2AzKVmiTg2eQr2d-rsuO9-E1IsiKWdB37jCF-c1baWZ2u4VJkVg1LgNDCqaj-pH-fRVfvmhgnfKg5hGUwXuConTbum-ZmWGoGlVbGvP-cIF-jJDECUE3BB7jauVMiSIycgnGvZduC5uAPeOkd-kGYxM4TkjoqATtKLQz8-P6SA_tmhBKQlTTjx-2pBboFadufQ%3D%3D&sa=X&ved=2ahUKEwiwpee23I-FAxXUamwGHVNoCQ0QmxMoAXoECFYQAw Cambridge **2020**, https://doi.org/10.17863/CAM.58311.
[46] Platform MaterialDigital, PMD Core Ontology Repository, **2023**, https://github.com/materialdigital/core-ontology/ (accessed: 03 March 2024).
[47] B. Bayerlein, M. Schilling, H. Birkholz, M. Jung, J. Waitelonis, L. Mädler, H. Sack, *Mater. Des.* **2024**, *237*, 112603.
[48] R. Moller, B. Neumann, M. Wessel, in *Proc. Integration of Speech and Image Understanding*, **1999**, pp. 101–115, https://doi.org/10.1109/ISIU.1999.824868.
[49] R. Cornet, A. Abu-Hanna, in *Artificial Intelligence in Medicine* (Eds: M. Dojat, E.T. Keravnou, P. Barahona), Springer Berlin Heidelberg, Berlin, Heidelberg, Germany **2003**, pp. 61–70, ISBN: 978-3-540-39907-0.
[50] C. J. H. Mann, *Kybernetes* **2003**, *32*, https://doi.org/10.1108/k.2003.06732iae.006.
[51] B. Konev, D. Walther, F. Wolter, in *Automated Reasoning* (Eds: A. Armando, P. Baumgartner, G. Dowek), Springer Berlin Heidelberg, Berlin, Heidelberg, Germany **2008**, pp. 259–274, https://doi.org/10.1007/978-3-540-71070-7_21.
[52] P. Warren, P. Mulholland, T. Collins, E. Motta, in *The Semantic Web: Trends and Challenges* (Eds: V. Presutti, C. d'Amato, F. Gandon, M. d'Aquin, S. Staab, A. Tordai), Springer International Publishing, Cham, Germany **2014**, pp. 550–564, https://doi.org/10.1007/978-3-319-07443-6_37.
[53] W3C, Shapes Constraint Language (SHACL), **2017**, https://www.w3.org/TR/shacl/ (accessed: 03 March 2024).
[54] LinkML, Linked Data Modelling Language, **2021**, https://linkml.io/ (accessed: 03 March 2024).
[55] github, GitHub, **2008**, https://github.com/ (accessed: 03 March 2024).
[56] Mat-o-Lab, MatPortal - the ontology repository for materials science, **2022**, https://matportal.org/, (accessed: 03 March 2024).
[57] F. J. Ekaputra, M. Sabou, E. Serral, E. Kiesling, S. Biffl, *Open J. Inf. Syst.* **2017**, *4*, 1.
[58] E. F. Aminu, I. O. Oyefolahan, M. B. Abdullahi, M. T. Salaudeen, *Int. J. Inf. Eng. Electron. Business* **2020**, *12*, 28.
[59] ISO/IEC 21383-1:2021-08, Information technology — Top-level ontologies (TLO) — Part 1: Requirements, International Organization for Standardization (ISO), Geneva, Switzerland, **2021**.
[60] ISO/IEC 21383-2:2021-11, Information technology — Top-level ontologies (TLO) — Part 2: Basic Formal Ontology (BFO), International Organization for Standardization (ISO), Geneva, Switzerland **2021**.
[61] Stanford University, Protégé, **2016**, https://protege.stanford.edu/ (accessed: 03 March 2024).
[62] Y. Chen, M. Schilling, P. von Hartrott, H. B. Nasrabadi, B. Skrotzki, J. Olbricht, *Integr. Mater. Manuf. Innov.* **2022**, https://doi.org/10.1007/s40192-022-00279-y.
[63] BAM Research, GitHub OntoPanel Frontend Repository, **2022**, https://github.com/BAMresearch/Ontopanel-frontend (accessed: 03 March 2024).
[64] JGraph Ltd, diagrams.net/draw.io, **2005**, https://www.diagrams.net/ (accessed: 03 March 2024).

[65] Mat-o-Lab, Materials Open Laboratory: Mat-O-Lab, **2021**, https://mat-o-lab.github.io/OrgSite/ (accessed: 03 March 2024).
[66] M. Poveda-Villalón, A. Gómez-Pérez, M. C. Suárez-Figueroa, *Int. J. Semant. Web Inf. Syst.* **2014**, *10*, 7.
[67] JSON Schema Org, json-schema-spec, **2022**, https://github.com/json-schema-org/json-schema-spec/ (accessed: 03 March 2024).
[68] FAIRsharing.org: QUDT; Quantities, Units, Dimensions and Types, https://doi.org/10.25504/FAIRsharing.d3pqw7, Last Edited: Friday, 6 May 2022, 11:03, Last Editor: Delphine Dauga, Last Accessed: Monday, 31 July 2023, 13:59.
[69] J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, C. Steinbeck, *Nucleic Acids Res.* **2016**, *44*, D1214.
[70] R. Gartner, in *Metadata*, Springer Link, Germany **2016**, pp. 1–13, https://doi.org/10.1007/978-3-319-40893-4_1.
[71] M. Schilling, S. Glaubitz, K. Matzak, B. Rehmer, B. Skrotzki, Full dataset of several mechanical tests on an S355 steel sheet as reference data for digital representations (1.0.0), Zenodo, **2022**, https://doi.org/10.5281/zenodo.6778335.
[72] DIN EN 10025-2, Hot rolled products of structural steels - Part 2: Technical delivery conditions for non-alloy structural steels; German version EN 10025-2:2019, Deutsches Institut für Normung (DIN), October **2019**.
[73] C. Boettiger, rdflib: A high level wrapper around the redland package for common rdf applications (Version 0.1.0), Zenodo, **2018**, https://doi.org/10.5281/zenodo.1098478.
[74] W3C, CSVW Namespace Vocabulary Terms, **2017**, https://www.w3.org/ns/csvw (accessed: 03 March 2024).
[75] W3C, RDF Data Access WG Charter, **2021**, https://www.w3.org/2003/12/swa/dawg-charter.html (accessed: 03 March 2024).
[76] Platform MaterialDigital, OntoDocker, **2022**, https://ontodocker.material-digital.de/, free registration required, (accessed: 03 March 2024).
[77] Pebbie GitHub Repositories, sparqlgpviz, **2020**, https://github.com/pebbie/sparqlgpviz/ (accessed: 03 March 2024).
[78] W3C, OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax, **2012**, https://www.w3.org/TR/owl2-syntax/#Class_Expression_Axioms/ (accessed: 03 March 2024).